Routing and Admission Control in General Topology Networks

Rainer Gawlick[*] *Anil Kamath*[†]

Serge Plotkin[‡] *K. G. Ramakrishnan*[§]

April 24, 1995

Abstract

Emerging high speed Broadband Integrated Services Digital Networks (B-ISDN) will carry traffic for services such as video-on-demand and video teleconferencing – that require resource reservation along the path on which the traffic is sent. As a result, such networks will need effective *admission control* algorithms.

The simplest approach is to use greedy admission control; in other words, accept every resource request that can be physically accommodated. However, in the context of symmetric loss networks (networks with a complete graph topology), non-greedy admission control has been shown to be more effective than greedy admission control.

This paper suggests a new *non-greedy* routing and admission control algorithm for *general topology* networks. In contrast to previous algorithms, our algorithm does not require advance knowledge of the traffic patterns. Our algorithm combines key ideas from a recently developed theoretical algorithm with a stochastic analysis developed in the context of reservation-based algorithms.

We evaluate the performance of our algorithm using extensive simulations on an existing commercial network topology and on variants of that topology. The simulations show that our algorithm outperforms greedy admission control over a broad range of network environments. The simulations also illuminate some important characteristics of our algorithm. For example, we characterize the importance of the implicit routing effects of the admission control part of our algorithm.

---

[*]MIT Laboratory for Computer Science, Cambridge, MA 02139. E-Mail: rgawlicklcs.mit.edu.

[†]Department of Computer Science, Stanford University, Stanford, CA 94305. E-Mail: kamath@cs.stanford.edu. Supported by NSF Grant CCR-9304971 and ARO Grant DAAH04-95-1-0121.

[‡]Department of Computer Science, Stanford University, Stanford, CA 94305. E-Mail: plotkin@cs.stanford.edu. Supported by NSF Grant CCR-9304971, ARO Grant DAAH04-95-1-0121, and Terman Fellowship.

[§]AT&T Bell Laboratories, Murray Hill, NJ 07974. E-Mail: kgr@research.att.com.

1 Introduction

Overview Future Broadband Integrated Services Digital Networks (B-ISDN) will carry a wide spectrum of new consumer services, such as video-on-demand, video teleconferencing, etc. A key characteristic of these services is that they require quality-of-service (QoS) guarantees. Assuring QoS requires reservation of resources. As a result, B-ISDN will likely allocate resources in terms of virtual circuits. In particular, creating a virtual circuit will require reservation of bandwidth on some path between the endpoints of the connection. Admission control algorithms are needed since network resources are limited.

Resources for virtual circuits are reserved using an admission control and routing algorithm. The admission control part decides which virtual circuits requests should be granted and which should be rejected; the routing part decides on the path used by the virtual circuit. The most natural approach is to make the admission control and routing decisions in a greedy manner. In other words, always route a circuit if there is a path with sufficient bandwidth; always use the minimum-hop path among the paths with sufficient bandwidth. Unfortunately, the greedy approach can lead to low throughput. For example, it will accept a virtual circuit request even if that request can only be accommodated along an excessively long path that might be more efficiently used by some future virtual circuits.

In this paper we introduce a new *non-greedy* admission control and routing algorithm for *general topology networks*. The goal of our algorithm is to maximize the number of virtual circuit requests that the network accepts, subject to the network's physical resource constraints. Simulations show that our algorithm significantly outperforms greedy algorithms for a wide range of network environments.

Our admission control and routing algorithm integrates several different approaches. We use the cost-benefit framework developed as part of the admission control algorithm in [4] and reinterpret it as a reservation scheme. We extend the techniques developed in the context of reservation-based algorithms [14, 15] and use these techniques to incorporate the stochastic properties of the offered traffic into the definition of the link-costs used in the cost-benefit framework.

Cost-benefit framework The following cost-benefit framework is introduced in [4]: Each link is assigned a cost that is *exponential* in its utilization. (The idea of using an exponential function to translate utilization into cost was proposed in [3].) The algorithm accepts a virtual circuit only if there exists a sufficiently cheap path i.e., a path where the sum of the link costs, integrated over the duration of the virtual circuit, is less than some predefined threshold. The accepted circuit is routed over one of these sufficiently cheap paths. The main parameters of this algorithm are the base of the exponent in the cost function and the value of the threshold.

The goal of [4] is to design an algorithm that achieves optimum performance in worst case scenarios. Thus, they evaluate their algorithm using the notion of a *competitive ratio* [16, 8]. The competitive ratio of an online virtual circuit routing and admission control algorithm is the maximum over all request sequences of the ratio of the number of requests accepted by the optimal algorithm for that sequence to the number of requests accepted by the online algorithm for the same sequence. An algorithm with a low competitive ratio is one that performs close to the optimal algorithm on all request sequences. Informally, the competitive ratio measures how much the performance of the online algorithm suffers in comparison to the optimal algorithm due to the fact that the online algorithm cannot predict future requests, since, for example, it does not know the traffic pattern. Competitive analysis does not make assumptions about the circuit requests, such as assumptions about the traffic pattern, and thus it provides a robust *worst-case* performance measure.

The algorithm in [4] achieves a competitive ratio of $O(\log LT)$, where L is the maximum number of links in a path and T is the ratio of the maximum to the minimum duration of a circuit. We will refer to this algorithm as the "AAP algorithm" in the rest of the paper. (Note that the above competitive ratio is too weak a performance guarantee to be useful in practice.)

EXP algorithm In this paper, we propose a new routing and admission control algorithm, which we refer to as the "EXP algorithm". Our algorithm design was guided by the following three general goals.

- *Use admission control to improve throughput performance.* There has been a consensus that non-greedy admission control can be used to improve the network throughput, measured in terms of the number of virtual circuits that the network accepts. This observation was first made in the context of symmetric loss networks [10]. More recently, this observation has been extended to general topology networks [14, 9, 15, 5]. Simulations of our algorithm confirm the performance advantages of non-greedy admission control.
- *Do not require advance knowledge of traffic patterns.* The traffic patterns for new B-ISDN networks will not be known in advance since the usage patterns for these networks are not well understood. Furthermore, the traffic patterns can vary dramatically over short periods of time. Hence, advance knowledge of the traffic pattern may be difficult to obtain. Our algorithm *does not require* advance knowledge of the traffic pattern. This design goal motivates both our use of concepts from a competitive analysis based algorithm and the manner in which we make use of the stochastic analysis. Our approach stands in sharp contrast to existing algorithms for general topology networks, which all require advance knowledge about the traffic patterns.
- *Minimize use of dynamic state information.* The use of detailed dynamic state information such as current link utilizations can significantly complicate the implementation of the algorithm in a distributed setting. Our algorithm seeks to minimize the use of that information by using static state information, e.g., number of links, to decide among the paths that meet the admission control criteria.

The AAP algorithm addresses two of our goals. It does not require advance knowledge of the traffic pattern, and it attempts to use admission control as a means to maximize the network throughput. Unfortunately, the AAP algorithm has several disadvantages that prevent it from being practical. First, the AAP algorithm deals only with admission control and does not address routing. Second, it requires that each circuit request specifies its duration. Third, each link must maintain and distribute large amounts of state information. Finally, the AAP algorithm is optimized for the worst-case situation and does not work well in common situations.

The EXP algorithm, proposed in this paper, retains some concepts from the AAP algorithm, e.g., the cost-benefit framework, but then modifies it to address AAP's shortcomings. Like the AAP algorithm, the EXP algorithm assigns each link a cost that is exponential in its utilization. A virtual circuit request is admitted if there is path whose cost is less than the value assigned to the circuit. We provide a routing component by routing on the minimum-hop path that meets the admission control criteria. To eliminate the need for knowing the circuit duration in advance, we eliminate AAP's integration of the link costs over the duration of the circuit. By eliminating the integration over the link costs and using a minimum-hop metric for routing, we substantially reduce that amount of state dynamic state information needed by the algorithm. Finally, to provide good performance in common situations, we use a novel mechanism for setting the parameters of the cost-benefit framework. The key parameters are the base of the exponential cost function and the value of a virtual circuit request. Our mechanism is an extension of the techniques developed in the context of reservation-based algorithms [14, 15]. Our simulations show that with our choice of parameters, the EXP algorithm performs consistently well over a wide range of network environments.

Related work Non-greedy admission control was first considered in the context of symmetric loss networks. Symmetric loss networks have a complete graph topology and equal capacity on each link. Furthermore, each source/destination pair has the same rate of circuit arrivals. Virtual circuits always use the direct link if it is available, otherwise they try to find an available path consisting of two links; paths consisting of more than two links are not considered. Admission control on direct paths is greedy. However, a two-link path is used only if both links have sufficiently low utilization. The advantage of using non-greedy admission control for the two-link paths in symmetric loss networks was first discussed in [10] and has since received extensive attention [1, 13, 12, 11]. An example of an admission control scheme based on the ideas developed in the context of symmetric loss networks is the Real Time Network Routing algorithm (ssc RTNR) [2] used in the AT&T long distance network.

Routing and admission control for general topology networks has also received attention. A cost based routing algorithm for general topology networks was developed by Ott and Krishnan [14]. Roughly speaking, their algorithm is based on the concept of costs that reflect the effect of routing and admission control decisions on the system performance. Unfortunately, their algorithm requires advance knowledge of the traffic pattern. Furthermore, their algorithm requires *complete* current state information for its path selection. An alternative approach was recently proposed by Sibal and DeSimone [15]. Their approach does not use cost functions. Rather, it is an extension of the ideas from symmetric loss networks to general topology networks. In their algorithm, the path selection is based on static criteria, with dynamic state information relevant only to the actual admission control decision. However, their admission control criteria still require advance knowledge of the traffic pattern.

Simulations. This paper provides an extensive set of simulations to evaluate the performance of our algorithm over a wide range of situations. The simulations are based on an existing commercial data network and some artificially generated networks. Among other things, we explore the effect of circuit bandwidths, circuit durations, and the degree to which the network load matches the network topology. The simulations also illuminate some important characteristics of our algorithm. For example, we characterize the effect of the implicit routing effects of the admission control part of our algorithm.

2 Admission Control via Cost Functions

This section introduces some of the the theoretical results in [4] and our new EXP algorithm.

2.1 Cost Benefit Framework with Exponential Costs

We represent the network by a capacitated (directed or undirected) graph $G(V, E, b)$. Let n denote the number of nodes in the graph. The capacity $b(e)$ assigned to each link $e \in E$ represents bandwidth initially available on this link.

The i^{th} virtual circuit request to AAP is a five-tuple consisting of the source node s_i, destination node d_i, starting time t_i^s, ending time t_i^f, and bandwidth requirement r_i. For simplicity, we assume that the routing is done at exactly time t_i^s. The algorithm either accepts the request, allocating bandwidth r_i along an appropriate route, or rejects the request. The goal of the algorithm is to maximize the total number of accepted requests[1]. Let $t_i = t_i^f - t_i^s$ denote the "holding time" of the circuit. Finally, let T denote the maximum possible holding time, t denote the minimum possible holding time, R denote the maximum possible requested bandwidth, and r the minimum possible requested bandwidth.

The routing decision is based on current information about the current and future *utilization* of the network links. Let P_i denote the route used to satisfy the ith request. The utilization of link e at time τ as seen by the routing algorithm when routing the kth circuit is defined as

[1] It is easy to modify the algorithm to optimize a general "profit" measure, where each routed request brings a predefined profit.

ROUTE$(s_j, d_j, t_j^s, t_j^f, r_j)$:
$\forall \tau, e \in E: \; c_e(\tau, j) \leftarrow \mu^{u_e(\tau,j)} - 1$;
if $\exists$ a path P in $G(V,E)$ from s_j to d_j s.t.
$$\int_{t_j^s \le \tau \le t_j^f} \sum_{e \in P} r_j c_e(\tau, j) \le \rho \text{ and } u_e(j) + r_j/b(e) < 1 \text{ for all } e \in P \qquad (*)$$
then route the requested virtual circuit on P, and set:
$\forall e \in P, t_j^s \le \tau \le t_j^f$,
$u_e(\tau, j+1) = u_e(\tau, j) + \frac{r_j}{b(e)}$
else reject the requested virtual circuit.

Figure 1: The AAP admission control algorithm.

follows:

$$u_e(\tau, k) = \sum_{\substack{e \in P_i, i<k, \\ \tau \in [t_i^s, t_i^f]}} \frac{r_i}{b(e)}.$$

After computing the utilization, the next step is to compute the *exponential cost*. The cost of link e at time τ as seen by the routing algorithm when routing the kth circuit is defined by

$$c_e(\tau, k) = b(e)(\mu^{u_e(\tau,k)} - 1),$$

where μ is a parameter.

The algorithm of [4] is shown in Figure 1. The main result of [4] is that choosing $\mu = 2nRT/r + 1$ and $\rho = nRT$ guarantees that for any sequence of requests where for all i we have $r_i \le \min_e\{b(e)/\log\mu\}$, the total number of requests accepted by the AAP algorithm is within $O(\log\mu) = O(\log(nRT/r))$ of the maximum total requests that can be routed by the best *offline* algorithm i.e., an algorithm that knows all the requests in advance and has infinite computational power. In fact, the analysis in [4] is easy to modify so that the performance bound improves to $O(\log(LRT/r))$ when the maximum allowed hop count is L.

2.2 EXP - A Practical Admission Control and Routing Algorithm

There are several aspects of the AAP algorithm that prevent it from being practical. First, the AAP algorithm deals only with admission control and does not address routing. Second, it requires a priori specification of duration for each request. Third, it requires each link to maintain and distribute large amounts of state information. Finally, the AAP algorithm is optimized for the worst-case situation and does not work well in common situations. Addressing each of these issues lead us to the EXP algorithm, shown in Figure 2.

AAP is essentially only an admission control algorithm. The only requirement on a chosen route is that it meets the admission control requirements given in the starred line of Figure 1.

ROUTE(s_j, d_j, r_j):
if $\exists$ minimum-hop path P in $G(V,E)$ from s_j to d_j s.t.

$$\sum_{e \in P} \mu^{u_e(j)} \leq \rho \text{ and } u_e(j) + r_j/b(e) < 1 \text{ for all } e \in P \qquad (*)$$

then route the requested virtual circuit on P, and set:

$$\forall e \in P, u_e(j+1) = u_e(j) + \frac{r_j}{b(e)}$$

else reject the requested virtual circuit.

Figure 2: The EXP admission control and routing algorithm.

Thus, AAP would permit choosing the longest path from among those meeting the admission control requirements. In contrast, EXP provides an explicit way to choose a route. Specifically, EXP chooses the minimum hop path that meets the admission control requirements. We make no claims about the optimality of this choice, but note the following advantages. A minimum hop path uses the fewest physical resources. Furthermore, a minimum hop path is determined by static rather than dynamic state information. This has advantages for distributed implementations of our algorithm (see Section 2.3). Section 3.6 provides simulation data that suggests that the use of a minimum hop path leads to good performance over a wide range of network environments.

In AAP, the cost of a path is determined in the starred line of Figure 1. The cost is given by an integral over the duration of the virtual circuit. This approach has two problems: the duration of each circuit must be known in advance, and each link must maintain the ending time and bandwidth of each virtual circuit. To address these problems, we simplify the cost function. In particular, instead of using $\int_\tau r_j(\mu^{u_e(\tau,j)} - 1)$ we use $r_j \mu^{u_e(j)}$, eliminating the integration step. [2] Furthermore, for the moment, we restrict attention to the case where the bandwidth of each virtual circuit is the same (denoted by r) and the capacity of each link is the same (denoted by b). As a result, r_j becomes a constant that gets absorbed into the constant ρ and hence not used in the description of the algorithm.

The fact that AAP is optimized for the worst-case situation reflects itself in its poor choice for the constants ρ and μ. To address this issue, we provide a new mechanism for choosing ρ and μ. First we set the value of ρ relative to μ. We observe that a path consisting of a single link provides the most efficient use of resources possible and therefore should always accept a circuit request. Since the cost of a single link path is at most μ, we set $\rho = \mu$. This ensures that lack of capacity is the only reason that the admission control procedure does not accept a virtual circuit along a path consisting of a single link.

We compute μ as a function of the *critical utilization* u^*, i.e. the utilization above which a link should be reserved for single link paths. Given u^*, it is easy to calculate μ as follows.

[2] Eliminating the integration step can be justified in the context of competitive analysis [7].

Recalling that $\rho = \mu$, we define μ such that

$$2\mu^{u^*} \geq \mu.$$

Using an equality and solving for μ we have: $\mu = 2^{1/(1-u^*)}$.

To calculate u^*, we borrow from the stochastic analysis in [14, 15]. Consider a single link that can accommodate b/r simultaneous circuits. Assume that circuit request arrivals are Poisson with rate λ and that the durations are exponentially distributed with mean 1. Assume further that there are currently j circuits using the link. Then the probability that accepting an additional circuit on the link will cause another future virtual circuit request to be rejected due to lack of capacity is given by:

$$\frac{B(b/r, \lambda)}{B(j, \lambda)}$$

where B is the standard b-erlang loss formula [14, 15]. Now consider a network consisting of two links in series. For simplicity we will assume that the departures on each link are independent[3]. This assumption has become standard in the literature. Let λ be the Poisson arrival rate of virtual circuits requiring a single link path. Assume that the two links currently both carry j circuits. Since a two link path could potentially block two single link paths, we require that the probability that a single link requested is rejected due to capacity constraints to be less than .5 for both links. In other words, two link path is rejected in favor of one link path once $\frac{B(b/r,\lambda)}{B(j,\lambda)} > .5$. The utilization, $u^* = jr/b$, for which two link path is rejected in this scenario can now be calculated if λ and b/r are given. This above analysis is similar in spirit to the analysis in [15].

The value of u^* depends on the values λ and b/r. The value for b/r is known as part of the network description. Determining the correct value for λ is more complicated. Above we define it as the arrival rate of single-link virtual circuit requests. Unfortunately, this arrival rate is highly dependent on the topology and traffic matrix of the network. Recall that one of our goals is to eliminate this dependence. Consequently, we propose the following heuristic for setting λ. Discussions with engineers charged with operations for several commercial networks suggest that 2% is the highest loss rate that a network should ever produce. We use this 2% figure to calibrate λ. In particular, we assume that the arrival rate of single-link circuits to any link is never more than λ^*, where λ^* is the arrival rate needed to generate a 2% loss rate on a single link in the absence of any other traffic. We set u^* using λ^*. By using λ^*, we are essentially calibrating our algorithm for the most aggressive admission control policy that will realistically be required[4]. In Section 2.3 we discuss why this aggressive form of admission control does not compromise performance in most situations. The simulations in Section 3.7 explore the sensitivity of EXP to λ^*.

[3]This is obviously an approximation since two link paths will create a dependency in the departure processes of the links.

[4]This is not strictly true when there is a large number of alternate short paths for a single link path. In particular, the stochastic properties that keep λ^* significantly below the capacity bound become less important.

2.3 Discussion

In defining EXP we have retained three key insights from the AAP algorithm. First, we have retained the cost-benefit framework for determining whether a circuit can be accepted on a particular path. The cost-benefit framework has the advantage that the use of a lightly loaded link does not penalize a circuit. As a comparison, consider the algorithm in [15], which establishes admission control criteria on a link-by-link basis. In particular, it rejects a virtual circuit request even if the admission control criteria fail on a single link of a path. Now consider a two-link (non primary) path with a highly utilized link and a lightly utilized link. The algorithm in [15] will reject a circuit along this path if the admission control criteria are not met on the highly utilized link. However, it might not be prudent to reject the circuit in this case.

The intuition is that the admission control algorithm should only protect scarce resources. Since the path in this example includes only a single scarce link it should be treated similar to a single link path using a scarce link. Recall that a single link path should always be accepted since it provides the most efficient use of resources possible. Our algorithm has the correct behavior in this case. Since we use a cost function that is exponential in the utilization, the highly utilized link will essentially be the only contributor to the cost of the path.

The second insight from the AAP algorithm that we retain is the relationship between admission control and the path length. Consider a path of length L where each link along the path has the same utilization. We now ask the following question: what is the maximum utilization u for which the L-link path should satisfy the admission control criteria? To answer this question in the context of an exponential function based algorithm we solve for u in the equation $\mu = L\mu^u$ to get $u = 1 - (\log L)/(\log \mu)$. Thus, the maximum utilization for which a path satisfies the admission control criteria decreases logarithmically with the length of the path.

Finally, we retain the observation that the admission control requirements provide essentially all of the state specific feedback that is needed for routing. By restricting the set of paths on which a circuit may be routed, the admission control component of EXP makes some implicit routing decisions. Once the state dependent restrictions are made, EXP can use state independent criteria (hop count) for deciding between the paths that meet the admission control restrictions. The ability to use state independent criteria has some advantages for distributed implementations of our algorithm. In particular, a distributed EXP algorithm can try paths in order of hop count. Each time it tries a path it can send a "setup" packet along the path to see it the path meets the admission control requirements. If it does, the path is chosen. If not, the next path is tried. (In practice, only few paths need to be tried before one can reject the circuit [6].) This approach is also used in [15]. We verify the sufficiency of using state independent criteria for deciding between the paths that meet the admission control restriction with the simulations in Section 3.6.

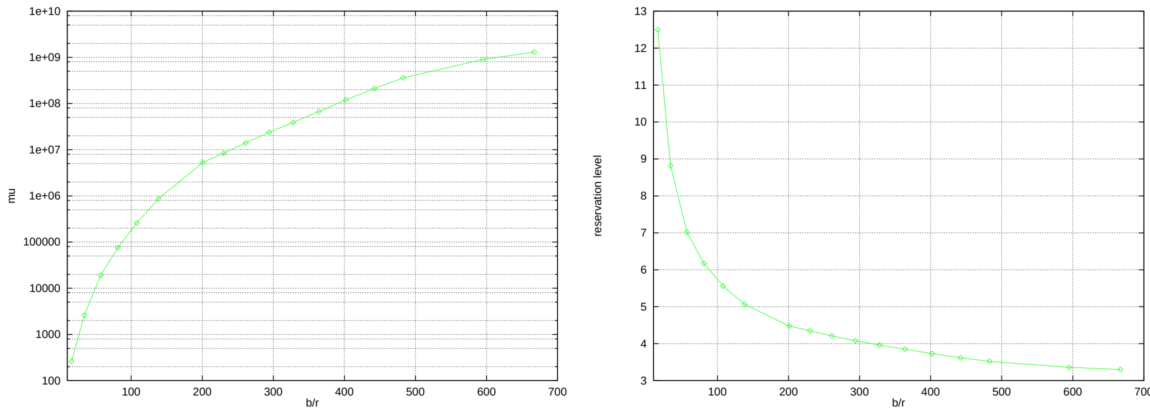

Figure 3: μ (left) and reservation (right) as a function of b/r, the number of circuits that fit on a single link.

Recall that our admission control algorithm is calibrated for very aggressive admission control since we assume that each link can reach a 2% rejection rate solely based on single link traffic. We provide an intuitive justification for this approach by considering two types of networks: one where the topology and the traffic matrix[5] are well matched and one where they are not well matched. In a network where the topology and the traffic matrix are well matched, there are direct links between source-destination pairs with large amounts of traffic. Thus the assumption that most links service primarily single link traffic is reasonable, especially at high loads. On the other hand, this assumption does not hold when the topology and the traffic matrix are not well matched. Thus, one might expect our admission control algorithm to be too aggressive. Fortunately this is not the case in practice. Since the network topology and the traffic matrix are not well matched, the load on the network links increases unevenly. Thus, while some links are heavily utilized other links still have low utilization. Therefore, the primary effect of the admission control algorithm is to cause circuits to use the lightly loaded links. In other words, the primary contribution of admission control is its effect on the routing decisions. The simulations described in Section 3.5 confirm this effect.

The constant μ for our admission control currently depends on only a single parameter: b/r, the number of circuits a link can simultaneously carry. We plot μ and the *reservation level* $1 - u^*$ as a function of b/r in Figure 3. The reservation level corresponds closely in spirit to the *trunk reservation level* of the symmetric loss network literature.

The dependence of the reservation level and μ on b/r raises two important issues: How do we handle situations where circuits have differing bandwidths? How do we deal with networks where the links have non-uniform capacity? We defer these issues to a subsequent paper. We note however, that the previously known approaches do not address these issues.

[5]The traffic matrix gives the percentage of the total network traffic that goes between each source-destination pair.

3 Simulation Results

This section evaluates the performance of the EXP algorithm against a greedy admission control strategy that uses minimum-hop routing. Our simulations are based on an existing commercial topology. The simulations provide considerable insight into behavior of our algorithm.

3.1 An Existing Commercial Topology

The existing commercial network consists of 25 nodes and 61 links. The topology is pictured in Figure 4. The capacities of the links are all chosen to be 155 Mbps, which corresponds to SONET OC-3 service. The virtual circuits all require 1 Mbps in both directions. When we take into account the overhead from the ATM headers, each link can accommodate 140 simultaneous virtual circuits. Calculations described in the previous section imply that we should use reservation level of 5% and $\mu = 9.4e5$ (see Figure 3). The holding times are exponentially distributed with a mean of 30 minutes. Virtual circuit requests arrive as a Poisson process. The traffic matrix corresponds to the actual current traffic on the network. We call this simulation scenario the *base case*.

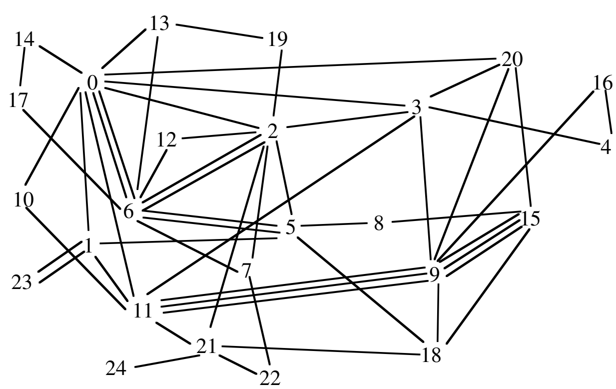

Figure 4: Topology of an existing commercial network.

In order to put the performance advantage of the EXP algorithm over the greedy strategy into perspective we wish to compute the performance of the optimum offline algorithm. Unfortunately, this computation is not tractable. Instead, we compute a lower bound on the optimum rejection rate by solving a multicommodity flow problem in which the objective function is to satisfy the maximum demand between node pairs without violating the capacity constraints, where the demand between node pairs is determined by the traffic matrix. In particular, the demand between nodes i and j is set to the average number of bits per second that are expected to be requested with i as the source and j as the destination. It is easy to see that the solution of this optimization problem is indeed a lower bound on the rejection rate. However, this lower bound may be far off from the true optimum since it does not take the stochastic properties of the circuit arrivals and departures into account. Furthermore, the multicommodity flow bound corresponds to the case where we are allowed to split a single virtual circuit over several paths.

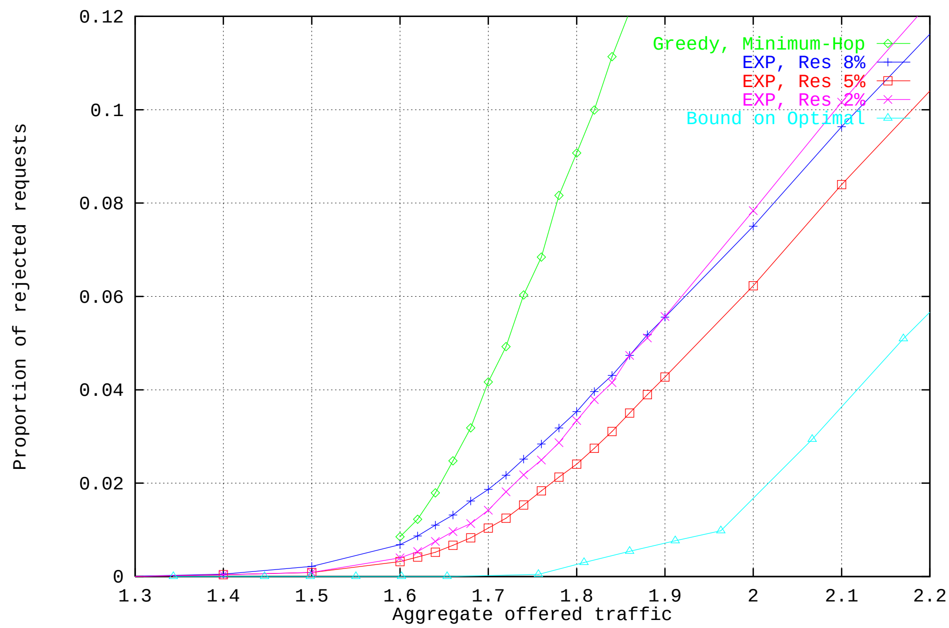

Figure 5: Simulation results for the commercial network.

Figure 5 compares the performance of the EXP algorithm with various reservation levels to the performance of a greedy minimum-hop algorithm and our lower bound on the performance of the optimum algorithm. The X-axis gives the (normalized) aggregate arrival rate in virtual circuits per second and the Y-axis gives the percentage of virtual circuits that are rejected. It can be seen that the EXP algorithm has a significant performance advantage over the greedy algorithm for a wide range of arrival rates. The EXP algorithm can maintain a much higher arrival rate given a target rejection (loss) rate. For a target maximum rejection rate of 2%, the EXP algorithm with the reservation level set at 5% ($\mu = 9.4e5$) can sustain an arrival rate that is approximately 8% higher than the arrival rate that can be sustained by the greedy algorithm. Taking our bound on the optimum algorithm as 100%, EXP achieves approximately 88% throughput, while the greedy algorithm achieves only 81%. We would like to reiterate that our bound on the optimum is quite optimistic and thus we believe that EXP achieves substantially more than 88% of the real optimum throughput.

The relative performance advantage of the EXP algorithm increases with arrival rate. For example, the improvement for target maximum rejection rate of 10% is 20%. In this case, EXP achieves approximately 88% of the bound on the optimum performance, while the greedy algorithm achieves only 73%.

Notice that the reservation level is a relatively forgiving parameter. In particular, Figure 5 also includes the results for reservation levels of 8% ($\mu = 5.8e3$) and 2% ($\mu = 1.1e15$). (The fact that the reservation level is a forgiving parameter was previously observed in the context of symmetric loss networks [11].)

3.2 Varying Virtual Circuit Bandwidth

A key factor in determining the correct reservation level and the correct value of μ is the number of virtual circuits that can be simultaneously accommodated on a single link. Figure 3 shows this effect analytically. We also illustrate this effect using simulations. The simulations are the same as in the base case except for the bandwidth of the virtual circuits. The right graph in Figure 6 corresponds to 200 Kbps circuits and the left graph corresponds to the case of 5 Mbps circuits. The optimal reservation levels and μ values for these cases are 10%, 989 and 3.3%, $1.6e9$, respectively. In each graph, we plot results for both values of μ. The simulations confirm that the reservation level should decrease as the number of simultaneous circuits that can be accommodated increases. Furthermore, the performance advantage of the EXP algorithm over the greedy algorithm increases with the number of circuits that can be simultaneously accommodated on a single link. In particular, in the case of 5 Mbps circuits, the EXP algorithm is 2% better than the greedy minimum-hop algorithm for a target maximum rejection rate of 2%, while for 200 Kbps circuits EXP is better by 9%. At a target maximum rejection rate of 4%, the improvements are 5% and 12% for the 5 Mbps and 200 Kbps cases, respectively.

The dependence of the correct reservation level on the number of circuits that can be simultaneously accommodated demonstrates the importance of incorporating stochastic properties into our analysis. An analysis based entirely on competitive analysis would not be able to predict this dependence.

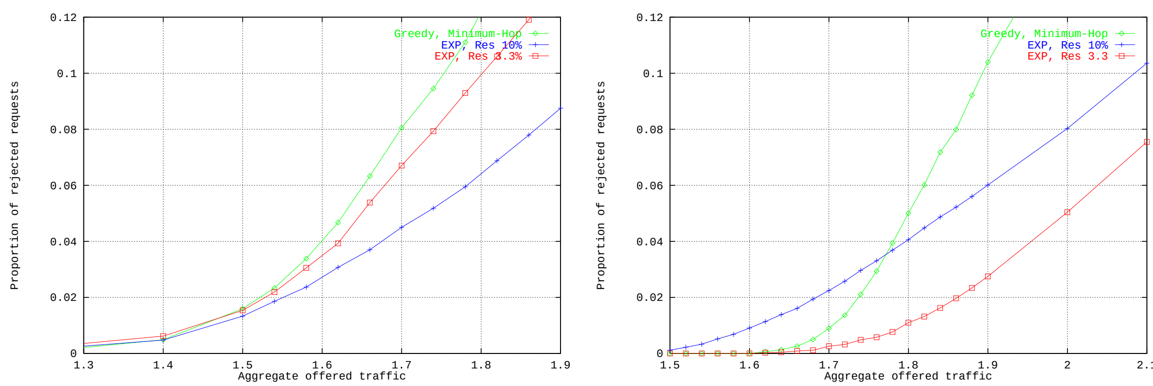

Figure 6: Simulations for virtual circuits with bandwidth of 5 Mbps (left) and 200 Kbps (right).

3.3 Varying Duration

The simulation results shown in Figure 7 use a *bimodal* distribution on the durations. This distribution tests the relative performance when there is a mix of short duration and long duration circuits. The duration of each circuit comes either from an exponential distribution with mean 6 minutes or from an exponential distribution with mean 30 minutes. Circuits are split between these two mean durations to ensure that each mean duration contributes approximately half of the currently active circuits. Figure 7 shows that there is no observable change in the relative performance of our EXP algorithm and greedy minimum-hop algorithm.

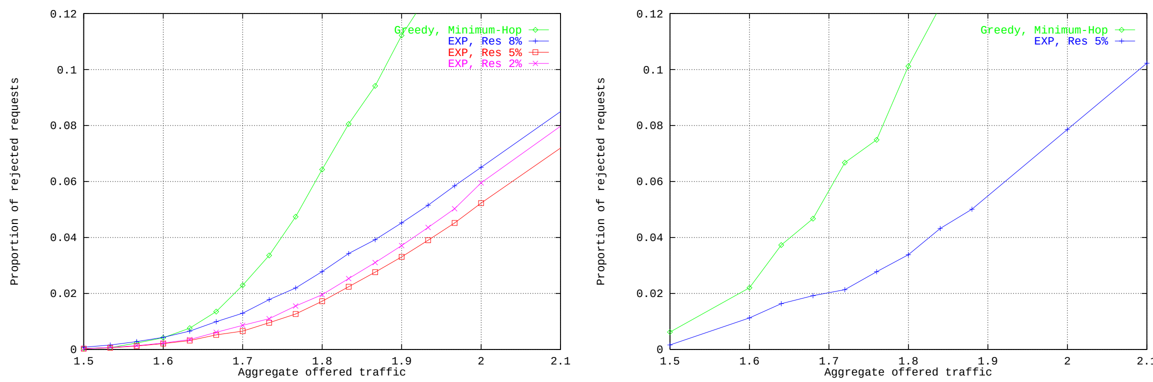

Figure 7: Simulations for virtual circuits with bimodal duration distribution (left). Simulations for virtual circuits with dynamic traffic patterns (right).

3.4 Dynamic Traffic Patterns

This section investigates the robustness of our algorithm to environments with very dynamic traffic patterns. In particular, the simulations in Figure 7 randomly change the traffic matrix at time intervals of one mean circuit duration. Each change to the traffic matrix alters the traffic between any source-destination pair to a value picked uniformly at random between 0 and twice its value in the base simulation scenario. The results show that the EXP algorithm maintains its performance advantage over the greedy admission control strategy.

3.5 The Routing Effects of Admission Control

Even though our EXP algorithm uses a static minimum-hop criterion to decide among paths, the EXP algorithm includes an implicit state dependent routing component. The implicit state dependent routing results from the restrictions that the admission control component of the algorithm places on the set of paths from which the minimum-hop routing component of our algorithm can choose. In this section we seek to quantify the relative contributions made by the implicit state dependent routing component of the EXP algorithm and the admission control component of the EXP algorithm. Quantifying the relative contributions will also give simulation-based support to the justification in given Section 2.3 for our aggressive approach to choosing the reservation level.

To quantify the routing effect of our EXP algorithm we study the performance of a new greedy admission control algorithm that makes routing decisions that are similar to those of the EXP algorithm. The new algorithm, called EXP-AC, is the same as the EXP algorithm except that it includes one additional step: If the EXP algorithm rejects the circuit, then the EXP-AC algorithm routes the circuit on the shortest path with respect to link costs given by $1 + \mu^{(u-1)}$, where u is the utilization of the link[6]. If, on the other hand, the EXP algorithm routes the circuit, then the EXP-AC algorithm uses the same path as the EXP algorithm.

The relative contributions made by the implicit routing and the admission control depend on the relationship between the topology and the traffic matrix. The simulations measure the percentage of the improvement over the greedy minimum-hop algorithm that is due to the implicit routing effects of the admission control (i.e., the percentage improvement achieved by the EXP-AC algorithm) as the degree to which the traffic matrix matches the topology changes.

Our simulations show that in the base case, the EXP-AC algorithm achieves 93% of the improvement that is achieved by the EXP algorithm. Thus, the implicit routing effects dominate. However, when the traffic matrix matches the topology perfectly, EXP-AC provides 0% of the improvement that is achieved by the EXP algorithm. Specifically, EXP-AC provides no improvement over the greedy minimum-hop algorithm while EXP provides over 10% improvement. In other words, the increase in throughput in this case is due to the admission control component of EXP. At the mid point of these extreme cases, (i.e., the traffic matrix is a 50%/50% linear combination of the base case and the traffic matrix that matches the topology perfectly) the EXP-AC algorithm achieves 46% of the improvement that is achieved by the EXP algorithm. In other words, in this case the implicit routing effects and the actual admission control contribute equally.

Now recall the discussion of Section 2.3. The following argument was used in Section 2.3 to justify the aggressive approach to to choosing the reservation level: If the traffic matrix does not match the topology then the main effect of the EXP algorithm will be through its effect on routing instead of through actual admission control. If the traffic matrix closely matches the topology, then the links are utilized in a uniform manner, which immediately justifies the aggressive reservation. The simulations support this argument.

While we have only investigated the relative contribution made by the implicit routing of the admission control criteria and actual admission control in the context of the EXP algorithm, we

[6] Since no path meets the admission control criteria, it is not clear how to route in such a way as to capture routing effects of the admission control. The weight function seeks to find a path that comes closest to meeting the admission control requirements, without choosing a path that is too long.

expect to find similar results for other admission control algorithm proposed in the literature.

3.6 Cost Based Routing vs Minimum Hop Routing

In order to facilitate a distributed implementation, our algorithm attempts to minimize its use of dynamic state information, such as link utilization. In particular, the algorithm uses a static minimum-hop metric to decide among the paths that meet the admission control criteria. The obvious alternative to using the minimum-hop metric is a metric based on the link utilization. For example, one could choose the minimum cost path with respect to the exponential cost metric used for admission control. This section describes simulation results that support our claim that there are no performance penalties for using a static minimum-hop metric to decide among the paths that meet the admission control criteria. In particular, the simulations show that the inherent routing effects of the admission control provide sufficient state depended information to the routing decision.

The simulations compare the performance of the EXP algorithm to a modified algorithm that we will refer to as "EXP-MC". EXP-MC chooses the minimum cost path in the exponential cost metric used for admission control. If that path satisfies the admission control criteria, i.e., the cost of the path is sufficiently low, EXP-MC accepts the circuit. Otherwise, EXP-MC rejects the circuit. The essential difference between EXP and EXP-MC is that EXP uses a static minimum-hop metric to decide between the paths that meet the admission control requirements, while EXP-MC uses a minimum cost metric that is based on link utilizations.

A key parameter in determining the relative performance of the algorithms is the degree to which the traffic matrix and the topology match. When the traffic matrix and the topology are well matched, we would actually expect the EXP algorithm to outperform the EXP-MC algorithm. In this case, most virtual circuit paths should consist of one link and thus EXP-MC's greater tendency to use multi-link paths harms its performance relative to that of EXP. On the other hand, when the traffic matrix and the topology are not well matched, we would expect EXP-MC to outperform EXP. The simulations show that the performance differences between EXP and EXP-MC are not great. When the traffic matrix and the topology are perfectly matched, EXP enjoys a 2% performance advantage over EXP-MC. In other words, the arrival rate at which EXP reaches a 2% circuit rejection rate is 2% larger than the arrival rate at which EXP-MC reaches a 2% circuit rejection rate. In the base case, where the traffic matrix and the topology are not well matched, EXP-MC enjoys a 2% performance advantage over EXP.

3.7 Varying Maximum Loss Rates

Recall that the value of μ depends on the maximum loss rate. In particular, we use the maximum loss rate to set λ^*, which we use to set μ (see Section 2.2). Based on discussions with engineers charged with the operations of several commercial networks, we use a maximum loss rate of

2%. Since this 2% value is somewhat arbitrary, we need to considered the sensitivity of EXP to this value. To test the sensitivity, we consider some extreme values for the maximum loss rate. In particular, consider a low value of .1% and a high value of 4%. In the base simulation, the low value of .1% leads to a reservation level of 2.7% while the high value of 4% leads to a reservation level of 7%. Examining Figure 5, we note that there are only small performance differences for reservation levels determined based on maximum rejection rates that are in the interval $[.1\%, 4\%]$. Hence, the performance our EXP algorithm is not very sensitive to the choice of maximum loss rate.

Acknowledgments

We would like to thank Debasis Mitra for his insight and support. We would also like to thank Flavio Bonomi, Bala Rajagopalan, Vikram Saksena and Deborah Estrin for helpful discussions.

References

[1] J. Akinpelu. The overload performance of engineered networks with nonhierarchical and hierarchical routing. *AT&T Bell Laboratories Technical Journal*, 63(7):1261–1281, September 1984.

[2] G.R Ash, J. Chen, A.E. Frey, B.D. Huang, C. Lee, and G.L. McDonald. Real-time network routing in the AT&T network-improved service quality at lower cost. In *Proceedings of the IEEE Global Telecommunications Conference, Orlando, Florida*, pages 115–136, December 1992.

[3] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and Orli Waarts. On-line load balancing with applications to machine scheduling and virtual circuit routing. In *Proceedings of the* 23^{rd} *Annual ACM Symposium on Theory of Computing, San Diego, California*, May 1993.

[4] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput competitive on-line routing. In 34^{th} *Annual Symposium on Foundations of Computer Science, Palo Alto, California*, November 1993.

[5] R. Gawlick, C. Kalmanek, and K. Ramakrishnan. On-line routing for permanent virtual circuits. In *Proceedings of IEEE Infocom*, April 1995.

[6] R. Gawlick, A. Kamath, S. Plotkin, and K. Ramakrishnan. Distributed routing and admission control. In preparation, 1995.

[7] R. Gawlick, A. Kamath, S. Plotkin, and K. Ramakrishnan. Routing and admission control using known distributions on holding times. In preparation, 1995.

[8] A. Karlin, M. Manasse, L. Rudolpoh, and D. Sleator. Competitive snoopy caching. *Algorithmica*, 3(1):79–119, 1988.

[9] F.P. Kelly. Routing in circuit-switched networks: optimization, shadow prices and decentralization.formal verification of parallel programs. *Advances in Applied Probability*, 20:112–144, 1988.

[10] R. Krupp. Stabilization of alternate routing networks. In *Proceedings of IEEE Internation Conference on Communications*, page 3I.2.1, 1982.

[11] D. Mitra, R. Gibbens, and B. Huang. State-dependent routing on symmetric loss netowkrs with trunk reservations – I. *IEEE Transactions on Communications*, 42(2), February 1993.

[12] D. Mitra and R. Gibbons. State dependent routing on symmetric loss networks with trunk reservations ii: Asymptotics, optimal design. *Annals of Operations Research*, 35(3):3–30, 1992.

[13] D. Mitra and J. Seery. Comparative evaluations of randomized and dynamic routing strategies for circuit switched networks. *Annals of Operations Research*, 39(1):102–116, 1991.

[14] T. Ott and K. Krishnan. State dependent routing of telephone traffic and the use of separable routing schemes. In *Proceedings of the 11th international teletraffic congress, Kyoto, Japan*, 1985.

[15] S. Sibal and A. DeSimone. Controlling alternate routing in general-mesh packet flow networks. *Proceedings of SIGCOMM 94*, 1994.

[16] D. Sleator and R. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.